Vendor Name: InfoSec Global
Product Name: AgileSec FIPS Module
Product Version: 1.0
Document Version: 1.1
FIPS 140-2 Non-Proprietary Security Policy
Level 1 Validation
December 11, 2018

InfoSec Global Inc.
www.infosecglobal.com

Author	Version	Date	Updates
InfoSec Global	1.0	July 2018	Initial version
InfoSec Global	1.0	December 2018	Updates

Contents

1. Introduction

1.1 Purpose

This is a non-proprietary FIPS 140-2 Security Policy for InfoSec Global's AgileSec SDK cryptographic module (AgileSec FIPS Module). It describes how this module meets all the FIPS 140-2 Level 1 requirements. This Policy forms a part of the submission package to the validating lab.

2. Security Level

AgileSec FIPS Module meets the Security Level 1 requirements of FIPS 140-2.

Table 2.1: Module Compliance Table

Security Requirements Section	Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles and Services and Authentication	1
Finite State Machine Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A
Overall Level of Certification	1

3. Product Description

3.1 AgileSec SDK and AgileSec FIPS Module

AgileSec SDK (software development kit) allows for seamless integration of cryptographic agility into software applications. This C-based cross-platform SDK implements the strongest internationally standardized cryptography and enables remote monitoring and upgrade of cryptographic algorithms in software and devices after their sale and deployment. It supports country-specific sovereign and custom cryptography, and new cryptographic algorithms.

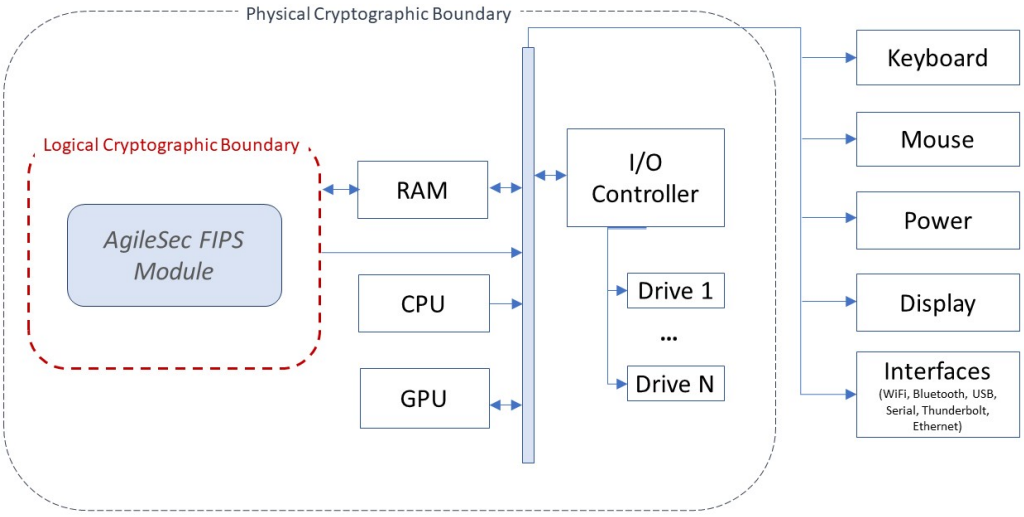
AgileSec FIPS Module is a component of AgileSec SDK. Its purpose is to provide secure and optimized implementations of FIPS 140-2 approved cryptographic algorithms.

3.2 Cryptographic Boundaries

The **logical cryptographic boundary** of AgileSec FIPS Module is the contiguous block of object code and data contained in the dynamic link library produced by compiling and linking the module source. The **physical cryptographic boundary** is the general purpose hardware executing the machine code of AgileSec FIPS Module. This hardware includes the central processing unit(s), main memory, system bus, and peripherals including storage drives, network cards, keyboard / consoles / mice, and attached displays.

The figure 3.1 presents the cryptographic boundaries of the module.

Figure 3.1: Cryptographic Boundaries of AgileSec FIPS Module



The module provides an Application Programming Interface (API) by which all cryptographic

functions are accessed. The module exposes the following API's:

Table 3.1: API's of AgileSec FIPS Module

API	Purpose
`isg_crypto_provider_entry`	AgileSec SDK dynamic entry point
`isg_FipsCryptoRegister`	AgileSec FIPS Module dynamic entry point

AgileSec SDK loads the FIPS-approved algorithms via one of these entry points, depending on how linking was done. Both of these go to the same internal entry point.

4. Module ports and interfaces

AgileSec FIPS Module is a multi-chip standalone software cryptographic module which operates within a commercially available general-purpose computing platform running on a commercially available operating system.

The module contains the following interfaces:

- **Data input interface**: input parameters to all API function calls by the Crypto-Officer or Crypto-User entities.
- **Data output interface**: output parameters from all API function calls that return data as arguments or return values from the Crypto-Officer or Crypto-User entities.
- **Control input interface**: control input parameters of API function calls used to configure or control the operation of the module.
- **Status output interface**: return codes from API's called by the Crypto-Officer and Crypto-User entities.
- **Power interface**: the crypto registration API's or entry points implicitly called when loading the module.

The module's API interface is mapped onto the FIPS 140-2 interfaces as follows:

Table 4.1: Mapping Physical and Logical Interfaces

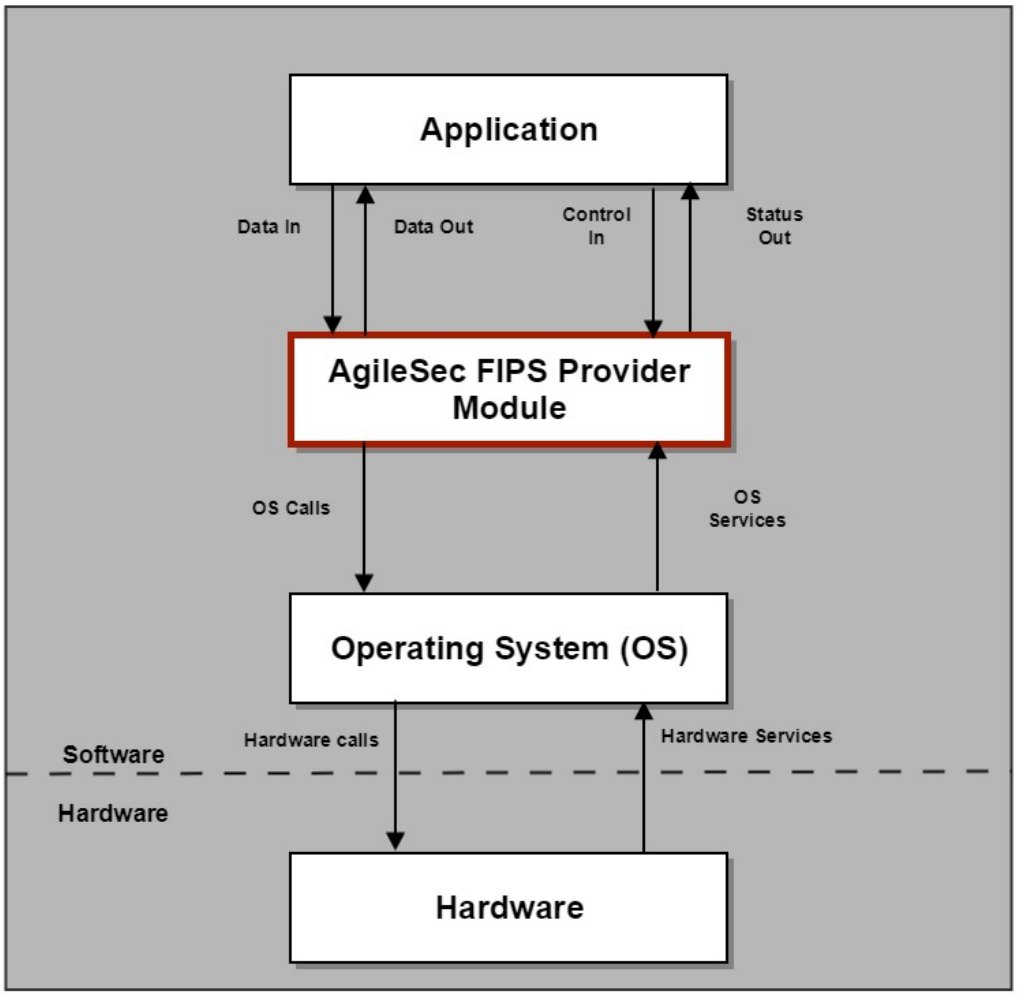
FIPS 140-2 Interface	Logical Interface	Physical Interface
Data Input	API calls	Ethernet port, USB ports, Serial ports, Thunderbolt ports, Bluetooth, WiFi
Data Output	API calls	Ethernet port, USB ports, Serial ports, Thunderbolt ports, Bluetooth, WiFi
Control Input	Control input parameters of API function calls that control the module's behavior	Keyboard and Mouse ports
Status Output	Status output parameters of API function calls that show the status of the module	Display Port

Table 4.1: Mapping Physical and Logical Interfaces

FIPS 140-2 Interface	Logical Interface	Physical Interface
Power Interface	Power interface	Power port of the tested hardware platform

The figure 4.1 presents the logical diagram of the module.

Figure 4.1: AgileSec FIPS Module Logical Diagram



5. Tested Configuration

This module is designed for commercially available general-purpose computer operating systems such as Linux, Windows, MacOS, and FreeBSD which provide a modifiable operational environment. This module is designed to operate in a single-user operational environment, where each user application runs in a virtually separated independent space. No claim can be made as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment which is not listed on the validation certificate. The module does not require any source code modifications to be recompiled and ported to another equivalent operational environment.

AgileSec FIPS Module has been tested on the following configurations:

- Qualcomm Snapdragon 800 @ 2.26GHz; LG Electronics Nexus 5X smartphone; FCC ID: ZNFD820 Android 8.1.0, android-ndk-r15c
- Intel Core i5 CPU @ 1.4GHz; Apple Mac Mini (Late 2014); FCC ID: QDS-BRCM 1069 OSX 10.12, 4 CPUs, clang/Apple LLVM version 8.1.0 (clang-802.0.42)
- Intel Xeon CPU ES-2630 v3 @ 2.4GHz; DELL PowerEdge R730xd; FCC ID: MSIP-REM-E2, VMware ESXi, 6.5.0, 7388607 Ubuntu 18.04, Linux x86, 4 CPUs, gcc 7.3.0
- Intel Xeon CPU ES-2630 v3 @ 2.4GHz; DELL PowerEdge R730xd; FCC ID: MSIP-REM-E2, VMware ESXi, 6.5.0, 7388607 FreeBSD 11.1-RELEASE-p1, amd64, 4 CPUs, clang 3.7.1
- Intel Xeon CPU ES-2630 v3 @ 2.4GHz; DELL PowerEdge R730xd; FCC ID: MSIP-REM-E2, VMware ESXi, 6.5.0, 7388607 Windows Server 2012 R2 Standard, 2 CPUs, mingw32-gcc (GCC) 6.4.0
- Freescale i.MX6 (ARMv7)CPU integrated in Toradex Colibri iMX6DL 512MB v1.1A HW; Radio module Texas Instruments WL1837MOD WiLink DB Wi-Fi and Bluetooth industrial module - FCC ID: Z64-WL18DBMOD, Linux colibri-imx6 4.1.41-2.7.3+g82f0f4f, arm-angstrom-linux-gnueabi-gcc (Linaro GCC 5.2-2015.11-2) 5.2.1 20151005

6. Secure Operation, Roles, Services, and Authentication

6.1 Secure Operation

An application can use AgileSec FIPS Module by:

- linking the .dll or .so file, or
- calling isg_CryptoProviderRegisterFromPath providing the path to the .dll or .so file.

Once the AgileSec FIPS Module is loaded, it executes a series of self-tests including an integrity test, self-tests, and pairwise consistency tests. If the self-tests are successful, the AgileSec FIPS Module is operational. The calling application will either:

- call isg_CryptoProviderRegisterFromPath, or
- link against the shared library and call isg_ProviderCreate followed by isg_FipsCryptoRegister.

If the integrity tests or other self-tests fail, an error is returned and no API calls into the AgileSec FIPS Module will succeed.

If the integrity tests and other self-tests succeed, no error is returned, the AgileSec FIPS Module is enabled so that API calls to it will succeed.

if any pairwise consistency test fails, an error is returned and no further API calls into the AgileSec FIPS Module will succeed.

If any of the integrity test, self-tests or pairwise consistency tests fail, then calls to any functions will return an error. The only way to recover is to reload the .dll or .so file.

6.2 Roles and Services

The AgileSec FIPS Module supports the Crypto-Officer and Crypto-User roles. The module does not support a maintenance role. The module does not support a bypass mode. The Crypto-User maps to the FIPs defined User role. Crypto-Officer and Crypto-User are responsible for restricting API calls to those marked as "FIPS 140-2 approved". The module does not specifically enforce the FIPS-approved and non-Approved modes but operates in a mixed mode of operation whereby the rules for the invocation of services and security functions are defined in this security policy. In the FIPS-approved mode of operation, all roles shall confine themselves to calling FIPS-approved algorithms, as marked in Table 6.1 and Table 7.1.

Table 6.1: Roles and Services in AgileSec FIPS Module

Service	Crypto Officer	Crypto User	Key/CSP
Controls			
Initialization	x	x	Does not access CSPs
Deinitialization	x	x	Does not access CSPs
Integrity test	x	x	HMAC-SHA-512 key for integrity test
Status	x	x	Does not access CSPs
Power-Up Self-Test	x	x	All keys that are generated are zeroized after the completion of the Self-Test
Symmetric ciphers			
Key generation	x	x	AES or Triple-DES keys
Key zeroization	x	x	All CSPs
Encrypt	x	x	AES or Triple-DES keys
Decrypt	x	x	AES or Triple-DES keys
Hash and Message Authentication			
Hash	x	x	HMAC Key
MAC	x	x	AES Key for CMAC with AES
Random number generation			
Instantiation	x	x	Does not access CSPs
Seeding	x	x	Seed
Request	x	x	Does not access CSPs
Uninstantiate (isg_ContextDestroy)	x	x	Does not access CSPs
Digital Signatures			
Keypair generation	x	x	DSA or ECDSA keys
Keypair zeroization	x	x	All CSPs
Signing	x	x	DSA, ECDSA and RSA keys
Verification	x	x	DSA, ECDSA and RSA keys
Key agreement			
Keypair generation	x	x	DH and ECDH keys
Keypair zeroization	x	x	DH and ECDH keys
Shared secret derivation	x	x	DH and ECDH keys
Key wrap			
Wrap-AES	x	x	AES keys
Unwrap-AES	x	x	AES keys
KDF			
PBKDF2[1]	x	x	Secret Value

6.3 Authentication

AgileSec FIPS Module does not support authentication.

[1]PBKDF2 is published in Internet Engineering Task Force Request for Comments (RFC) 2898 and maps to PBKDF defined in NIST SP 800-132. PBKDF2 should only be used for storage applications.

6.4 Security Rules and Guidance

6.4.1 GCM IV

In the FIPS-approved mode, when a GCM IV is generated randomly, the module enforces the use of a FIPS-approved DRBG in line with Section 8.2.2 of SP 800-38D:

- GCM IV generation uses a FIPS-approved DRBG and
- the DRBG seed is generated inside the module's physical cryptographic boundary.

Crypto-Officer and Crypto-User must ensure that:

- The GCM IV length is at least 96 bits (per SP 800-38D).

In the FIPS-approved mode, importing a GCM IV is non-conformant.

In line with Section 2.1 of IG A.5, in the event module power is lost and restored, the consuming application must ensure that any of its AES-GCM keys used for encryption or decryption are re-distributed.

6.4.2 Triple-DES

Crypto-Officer and Crypto-User must ensure that the module is compliant with SP 800-67 Rev. 2, namely that there are no more than 2^{20} 64-bit data block encryptions with the same Triple-DES key.

The use of two-key Triple-DES is disallowed for encryption, but can be used for decryption for legacy usage after 2015.

6.4.3 PBKDF2

In line with the requirements for SP 800-132, keys generated using the approved PBKDF must only be used for storage applications. Any other use of the approved PBKDF is non-conformant.

In the FIPS-approved mode, passwords must encode to at least 14 bytes (112 bits) and the salt must be at least 16 bytes (128 bits) long. The iteration count associated with the PBKDF should be as large as practical. For security considerations and further information on password, salt, and iteration count selection, consult Appendix A "Security Considerations" of SP 800-132.

7. Approved and Allowed Cryptographic Algorithms

AgileSec FIPS Module supports a wide range of cryptographic algorithms. Table 7.1 lists the approved algorithms in the FIPS-approved mode of operation.

Table 7.1: Cryptographic Algorithms in FIPS-Approved Mode

Algorithm	Specification	Options	Certificate
Symmetric ciphers			
AES	FIPS 197	128/192/256, ECB, CBC, CTR (External Counter Source), Encrypt, Decrypt	5534
AES GCM	FIPS 800-38D	128/192/256 using an internal IV of at least 96 bits, Encrypt, Decrypt	5534
Triple-DES	FIPS 800-67 Rev2	3-Key, ECB, CBC, CTR (External Counter Source), Encrypt, Decrypt	2787
Hash algorithms			
SHA1	FIPS 180-4	SHA-1 is disallowed only for digital signature generation (unless SP 800-52 TLS exception is met), digital signature verification using SHA-1 is still allowed.	4441
SHA2	FIPS 180-4	224/256/384/512	4441
SHA3	FIPS 202	224/256/384/512	50
HMAC / CMAC			
HMAC	FIPS 198-1	SHA1, SHA2 (224,256,384,512)	3686
CMAC	FIPS 800-38B	AES (128/192/256)	5534
Digital signatures			
ECDSA	FIPS 186-4	Key Pair Generation (P-224/P-256/P-384/P-521), Public Key Validation (P-224/P-256/P-384/P-521), Signature Generation and Verification ((P-224/P-256/P-384/P-521 and SHA1(for Verification only), SHA2 (224/256/384/512))	1489

Table 7.1: Cryptographic Algorithms in FIPS-Approved Mode

Algorithm	Specification	Options	Certificate
RSA PKCS1V15	FIPS 186-4	SigGenPKCS1.5 2048, 3072 with SHA2 (224/256/384/512) and 4096 with SHA2 (224/256/384/512) SigVerPKCS1.5 2048, 3072 with SHA1, SHA2 (224/256/384/512)	2968
RSA PSS	FIPS 186-4	SigGenPSS 2048, 3072 with SHA2 (224/256/384/512) and 4096 with SHA2 (224/256/384/512), SigVerPSS 1024, 2048, 3072 with SHA1, SHA2 (224/256/384/512)	2968
DSA	FIPS 186-4	Key Pair Gen (2048/3072), Sig Gen (2048, 3072 with SHA2 (224/256/384/512)) Sig Ver (1024, 2048, 3072 with SHA1, SHA2 (224/256/384/512))	1419
Key Agreement			
ECC CDH (CVL)	NIST SP 800-56A	KAS ECC (Full Unified, Ephemeral Unified, One Pass Unified, One Pass DH, Static Unified, Sect. 5.7.1.2 ECC CDH (CVL)) Must use curves P-224/P-256/P-384/P-521	1977
ECDH	NIST SP 800-56A	KAS ECC (Full Unified, Ephemeral Unified, One Pass Unified, One Pass DH, Static Unified, Sect. 5.7.1.2 ECC CDH (CVL)) Must use curves P-224/P-256/P-384/P-521	189
DH	NIST SP 800-56A	KAS FCC (dhEphem, dhStatic), must use Keysizes greater than 2048	189
Key Wrap			
AES Key Wrap	NIST SP 800-38F	Modes: Decrypt, Encrypt Key Lengths: 128, 192, 256 (bits) Plain Text Lengths: multiples of 64 bits	5534
pRNG			
HMAC DRBG	NIST SP 800-90A	Prediction Resistence Modes: Not Enabled, Modes: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	2192
PBKDF			

Table 7.1: Cryptographic Algorithms in FIPS-Approved Mode

Algorithm	Specification	Options	Certificate
PBKDF2	RFC 2898	Options: PBKDF with Option 1a. Functions: HMAC-based KDF using SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	Vendor Affirmed IG D.6
CKG			
CKG	NIST SP 800-133	Cryptographic Key Generation	Vendor Affirmed IG D.12

AgileSec FIPS Module includes some algorithms and associated services which are allowed in FIPS-approved mode of operation but are not approved. These algorithms are presented in Table 7.2.

Table 7.2: Allowed but not Approved Cryptographic Algorithms in FIPS-Approved Mode

Algorithm	Specification	Options
Key Agreement		
ECDH	NIST SP 800-56A	Must use curves P-224/P-256/P-384/P-521
DH	NIST SP 800-56A	Must use Keysizes greater than 2048
RNG		
NDRNG		The module obtains the entropy data from NDRNG to seed the DRBG. Note: NDRNG is outside of the module boundary.

AgileSec FIPS Module includes non-compliant algorithms and associated services, which are not allowed in the FIPS-Approved mode of operation. Their use will result in the module operating in a non-Approved mode. The CO shall zeroize keys when transitioning between Approved and non-Approved services and vice versa. If any non-compliant algorithms are used with any of the approved services (as listed in Table 6.1) then this will result in non-approved mode of operation.The list of non-approved algorithms is presented in Table 7.3.

Table 7.3: Non-Compliant Cryptographic Algorithms in AgileSec FIPS Module

Algorithm	Specification	Options
Asymmetric ciphers		
RSA OAEP	SP 800-56B	All options
RSA Raw		All options
RSA Key Generation		
Symmetric ciphers		
AES GCM with an external IV or an internal IV of less than 96 bits		
Key Agreements		
DH	NIST SP 800-56A	For keysizes less than 2048 providing less than 112 bits of security
ECDH	NIST SP 800-56A	For Curves: Brainpool (256R1, 384R1, 512R1) Curve25519, E25519, E25519COMPAT, E448
Digital signatures		
DSA	FIPS 186-4	For security strength less than 112-bits

Table 7.3: Non-Compliant Cryptographic Algorithms in AgileSec FIPS Module

Algorithm	Specification	Options
ECDSA	FIPS 186-4	Curves: Brainpool (256R1, 384R1, 512R1) Curve25519, E25519, E25519COMPAT, E448
EDDSA	RFC 8032	all options
RSA PKCS1V15	FIPS 186-4	Signature Generation (2048, 3072 with SHA3-224, SHA3-256, SHA3-384, SHA3-512 and 4096 with SHA1, SHA3-256, SHA3-384, SHA3-512) Signature Verification (1024, 2048 with SHA3-224, SHA3-256, SHA3-384, SHA3-512 and 3072 with SHA1, SHA3-224, SHA3-256, SHA3-384, SHA3-512)
RSA PSS	FIPS 186-4	Signature Generation (2048, 3072 with SHA3-224, SHA3-256, SHA3-384, SHA3-512 and 4096 with SHA1, SHA3-256, SHA3-384, SHA3-512) Signature Verification (1024, 2048 with SHA3-224, SHA3-256, SHA3-384, SHA3-512 and 3072 with SHA1, SHA3-224, SHA3-256, SHA3-384, SHA3-512)
pRNG		
HMAC DRBG	NIST SP 800-90A	SHA3-224, SHA3-256, SHA3-384, SHA3-512

Table 7.4 summarizes keys and CSPs used in the FIPS-approved mode.

Table 7.4: Key and CSP, Key Size, Security Strength, and Access

Algorithm	Key and CSP	Key Size	Strength	Access
AES	key	128-256 bits	128-256 bits	Create, Read, Use
AES GCM	key	128-256 bits	128-256 bits	Create, Read, Use
AES GCM	IV	128-256 bits	128-256 bits	Create, Use
Triple-DES	key	168 bits	112 bit	Create, Read, Use
HMAC	key	160-512 bits	128-256 bits	Use

Table 7.4: Key and CSP, Key Size, Security Strength, and Access

Algorithm	Key and CSP	Key Size	Strength	Access
CMAC AES	key	128-256	128-256 bits	Use
DSA	public key and private key	Public: 2048-15360 bits, Private: 224-512 bits	112-256 bits	Create, Read, Use
ECDSA	public key and private key	224-521 bits	112-256 bits	Create, Read, Use
RSA Signature	public key and private key	2048-15360 bits	112-256 bits	Create, Read, Use
DH	static/ephemeral public key and private key	Public: 2048-15360 bits, Private: 224-512	112-256 bits	Create, Read, Use
ECDH	static/ephemeral public key and private key	224-521 bits	112-256 bits	Create, Read, Use
PBKDF2	Secret value used in construction of Keyed-Hash key for the specified PRF.	Calling Application should pass in a Secret value of at least 14 bytes	112 bits	Use
PBKDF2	DPK	Data protection key	112 bits	Use
DRBG	entropy input string	entropy	length dependent on security strength	Use
DRBG	Seed	Seed	length dependent on security strength	Use
DRBG	V	V value	length dependent on security strength	Use
DRBG	key value	key	length dependent on security strength	Use
AES wrapping	key	key	128-256 bits	Use

Table 7.4: Key and CSP, Key Size, Security Strength, and Access

Algorithm	Key and CSP	Key Size	Strength	Access

Table 7.5 summarizes the methods for generation/input, storage, zeroization, and use of the keys and CSPs in the FIPS-approved mode.

Table 7.5: Key/CSP, Generation/Input, Output, Storage, Zeroization and Use

Algorithm	Key and CSP	Generation or Input	Output	Storage	Zeroization	Use
AES	key	Call isg_SymcipherKeySet or isg_SymcipherKeyGen	Call isg_SymcipherKeyGet	RAM	Call isg_SymcipherKeyDestroy	Encryption and Decryption
AES GCM	key	Call isg_SymcipherKeySet or isg_SymcipherKeyGen	Call isg_SymcipherKeyGet	RAM	Call isg_SymcipherKeyDestroy	Encryption and Decryption
AES GCM	IV	Call isg_AuthEncBegin	No output method available	RAM	Zeroization is done without user intervention	Initialization Vector
Triple-DES	key	Call isg_SymcipherKeySet or isg_SymcipherKeyGen	Call isg_SymcipherKeyGet	RAM	Call isg_SymcipherKeyDestroy	Encryption and Decryption
HMAC	key	Call isg_MacKeySet	No output method available	RAM	Call isg_MacKeyDestroy	Creating Message Authentication Code
CMAC AES	key	Call isg_MacKeySet	No output method available	RAM	Call isg_MacKeyDestroy	Creating Message Authentication Code
DSA	public key and private key	Call isg_KeyPairSet or isg_KeyPairGen	isg_KeyPairGet	RAM	Call isg_KeyPairDestroy	Signing and verifying
ECDSA	public key and private key	Call isg_EccKeyPairSet or isg_KeyPairGen	Call isg_KeyPairGet	RAM	Call isg_KeyPairDestroy	Signing and verifying
RSA Signature	public key and private key	Call isg_RSAKeyPairSet	Call isg_RSAKeyPairGet	RAM	Call isg_KeyPairDestroy	Signing and verifying

Table 7.5: Key/CSP, Generation/Input, Output, Storage, Zeroization and Use

Algorithm	Key and CSP	Generation or Input	Output	Storage	Zeroization	Use
DH	static or ephemeral public key and private key	Call isg_DHKeyPairGen or isg_DHKeyPairSet	Call isg_DHKeyPairGet	RAM	Call isg_KeyPairDestroy	Key Exchange
ECCDH	static/ ephemeral public key and private key	Call isg_EccKeyPairSet or isg_KeyPairGen	Call isg_EccKeyPairGet	RAM	Call isg_KeyPairDestroy	Key Exchange
PBKDF2	The password (key) is passed in to the function.	Input to function	None	RAM	The PBKDF2 function zeroizes intermediate values. It is the calling applications responsibility to zeroize the password.	The password that is passed into the function is used to derive a key.
PBKDF2	DPK	None	None	RAM	It is the calling applications responsibility to zeroize the DPK	The password that is passed into the function is used to derive a key.
DBRG	entropy input string	Input to function	None	RAM	zeroized by the function	Input entropy.
DBRG	seed	Input to function	None	RAM	zeroized by the function	Input entropy.
DRBG	V	None	None	RAM	call to isg_ContextDestroy	V value for DRBG function
DRBG	key value	None	Output from the function	RAM	isg_ContextDestroy	Key created by DRGB function.
AES Key Wrap	Key	None	Input to the function	RAM	It is the calling applications responsibility to zeroize the key.	The wrapping key that is used to wrap the key provided.

8. Module state machine

The AgileSec FIPS Module state machine uses the following states to ensure no cryptographic operations are performed unless the required self-tests have completed:

- Uninitialized
- Integrity check
- Self-test
- Enabled
- Crypto-Officer/User
- Error

The module starts in the **Uninitialized** state, transitioning immediately to the **Integrity Check** state upon module load. While in the **Integrity Check** and **Self-Test** state, all data output interfaces are disabled.

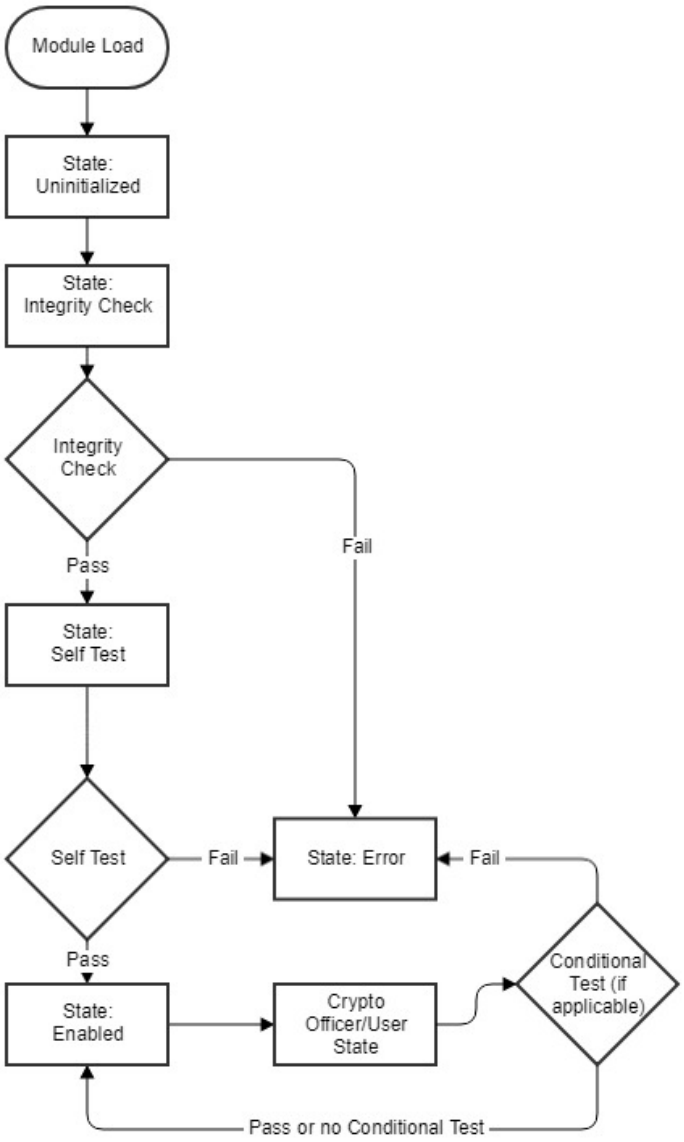
If the integrity check passes, the module transitions from the **Integrity Check** state to the **Self-Test** state. If the integrity check fails, the module transitions to the **Error** state.

If the self-test passes, the module transitions to the **Enabled** state. If the self-test fails, the module transitions to the **Error** state.

If Crypto-Officer or Crypto-User initiates a cryptographic operation, the module transitions from the **Enabled** state to the **Crypto-Officer/User** state. If the cryptographic operation requires a conditional test, then the conditional test is performed. If the conditional test fails, the module transitions to the **Error** state. If the conditional test passes or if no conditional test is required, the module transitions back to the **Enabled** state.

The only way to transition out of the **Error** state is to reload the module.

Figure 8.1: AgileSec FIPS Module State Diagram



9. Key Management

AgileSec FIPS Module provides the underlying functions to support key management according to FIPS 140-2 Level 1 requirements. Crypto-Officer and Crypto-User are responsible for selection of FIPS 140-2 approved algorithms (see Table 7.1). Crypto-User is also responsible for handling keys as required by FIPS 140-2.

9.1 Key Generation

AgileSec FIPS Module provides FIPS 140-2 compliant key generation. The underlying random number generation uses a NIST SP 800-90A compliant mode of HMAC DRBG without post-processing. The key generation methods implemented in the AgileSec FIPS Module for FIPS-approved services in FIPS-approved mode are compliant with SP 800-133. The module requests a minimum of 128 bits of entropy for each GET function call.

AgileSec FIPS Module uses a Non-Deterministic Random Number Generator (NDRNG) as the entropy source for seeding the DRBG. The NDRNG is provided by the operating system and is located in the operational environment, which is within the module's physical cryptographic boundary but outside of the module's logical cryptographic boundary

CAVEAT: The strength of the cryptographic keys generated by AgileSec FIPS Module is modified by the available entropy.

9.2 Key Establishment

AgileSec FIPS Module provides the following FIPS-approved or allowed key establishment techniques:

Diffie-Hellman: key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength.

EC Diffie-Hellman: key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength.

KTS: AES Cert. 5534; key establishment methodology provides between 128 and 256 bits of encryption strength.

9.3 Key Entry and Output

FIPS-approved algorithms must be used to encrypt keys exported from beyond the module's cryptographic boundary. Similarly, encrypted keys imported from beyond the module's cryptographic boundary must be encrypted with FIPS-approved algorithms.

9.4 Key Storage

As AgileSec FIPS Module is a low-level cryptographic toolkit, there is no persistent storage of keys/CSPs. AgileSec FIPS Module stores keys/CSPs in the volatile memory during its operation.

9.5 Zeroization of Keys

AgileSec FIPS Module provides zeroization functions (see Table 6.1). Zeroization of keys and CSPs, when they are no longer needed, must be performed by calling a destroy function of the respective object: otherwise AgileSec FIPS Module will not be functional.

10. Self-tests

10.1 Power-On Self-Tests

AgileSec FIPS Module performs the following power-up self-tests on initialization:

Table 10.1: Power-Up Self-tests performed by AgileSec FIPS Module

Algorithm	Type	Attributes
Integrity Test	KAT	HMAC-SHA512
AES KW	KAT	Seperate tests for wrapping and unwrapping
DH	KAT	Per IG 9.6
HMAC-SHA1	KAT	Satisfies SHA1 KAT per IG 9.1
HMAC-SHA224	KAT	Satisfies SHA224 KAT per IG 9.1
HMAC-SHA256	KAT	Satisfies SHA256 KAT per IG 9.1
HMAC-SHA384	KAT	Satisfies SHA384 KAT per IG 9.1
HMAC-SHA512	KAT	Satisfies SHA512 KAT per IG 9.3
HMAC-SHA3-224	KAT	Satisfies HMAC-SHA3-224 KAT and SHA3-224 per IG A.11
HMAC-SHA3-256	KAT	Satisfies SHA3-256 KAT per IG 9.1
HMAC-SHA3-384	KAT	Satisfies HMAC-SHA3-384 KAT and SHA3-384 KAT per IG A.11
HMAC-SHA3-512	KAT	Satisfies HMAC-SHA3-512 KAT and SHA3-512 KAT per IG A.11
CMAC-AES128	KAT	Generate
CMAC-AES192	KAT	Generate
CMAC-AES256	KAT	Generate
AES128 ECB	KAT	Seperate tests for encrypt and decrypt
AES192 GCM	KAT	Seperate tests for encrypt and decrypt
AES256 GCM	KAT	Seperate tests for encrypt and decrypt
Triple-DES ECB	KAT	Seperate tests for encrypt and decrypt
RSA	KAT	Sign/Verify with PKCS1V15
DSA	PCT	Sign/Verify
ECDSA	PCT	Sign/Verify with P-256 and SHA256
ECDH CDH	KAT	Shared secret derivation with P-256 per SP 800-56A, IG 9.6
DRBG	KAT	HMAC DRBG: SHA256. DRBG Health Tests (Instantiate, Generate, Reseed)

AgileSec FIPS Module is initialized automatically when a shared library, or a .dll library (depending on the platform) is loaded. The library can be loaded by:

- a dynamic call to `dlopen()` or to `LoadLibrary()` on Windows;
- executing an application which is linked to the shared library.

The entry point is:

void

isg_fips_entry(void) __attribute__((constructor));

In each instance, the module executes the power-up self-tests with no operator intervention by calling `isg_FipsCryptoIntegrityCheck()` followed by `isg_FipsCryptoSelfTests()`. If either of those functions returns non-zero, the module state is set to `FIPS_STATE_ERROR`, and no further cryptographic operations are permitted. If the self-tests complete successfully, the module state will be set to `FIPS_STATE_ENABLED` and cryptographic operations will be processed normally.

The power-up self-tests may be executed on demand by calling `isg_FipsCryptoIntegrityCheck()` and `isg_FipsCryptoSelfTests()`. Each of those functions return 0 upon success, or a non-zero exit code upon failure. The return code can be interpreted from the set contained in the header file `isg_return.h`.

AgileSec FIPS Module software integrity test deploys HMAC SHA512 validation to verify the integrity of the module.

10.2 Conditional Self-Tests

The following conditional self-tests are run:

- DSA key generation - Pairwise Consistency Test
- DH key generation - Pairwise Consistency Test
- ECDH key generation - Pairwise Consistency Test
- ECDSA key generation - Pairwise Consistency Test
- Continuous test on NDRNG (entropy source) 64 bit blocks

10.3 Critical Function Tests

The following critical function tests are applicable to the HMAC DRBG, as per Section 11 of SP 800-90A, and are triggered at power-up and when a reseed is called.

- Instantiate Test
- Generate Test
- Reseed Test

11. Mitigation of Other Attacks

AgileSec FIPS Module does not claim to mitigate any attacks.

Copyright and Licensing

[1] InfoSec Global. AgileSec FIPS Module Software Security Policy, 2018.